Advanced Features of the Shift Automated File Transfer Tool

April 10th, 2013

NASA Advanced Supercomputing Division

Overview

- Shift summary
  - What is it?
  - Why should you use it?
- Review of basic usage
  - Initiating transfers
  - Managing transfers
  - Monitoring transfers
- New features and advanced usage
  - Local and remote tar creation/extraction
  - Transfer parallelization using multiple clients/hosts
  - Rsync-like synchronization
  - ...more

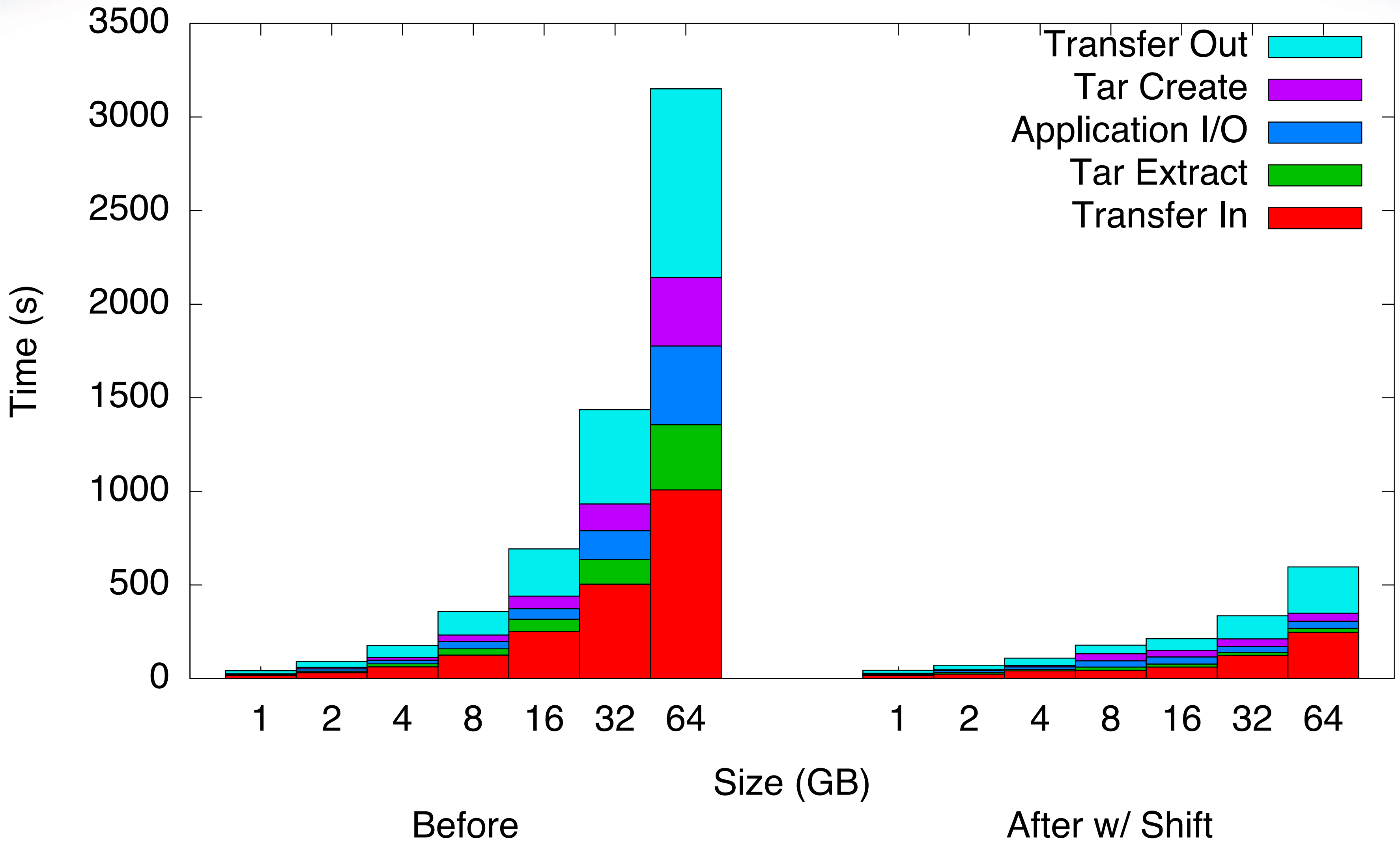
Question? Use the Webex chat facility to ask the Host

Self-Healing Independent File Transfer (Shift)

- What is it?
  - Automated file transfer mechanism
  - Supports local, intra-enclave, and remote transfers
  - Has facilitated transfers of >1.4 PB of data since April 2012
- Why should you use it?
  - Unified interface with simple cp/scp syntax
  - Fire and forget transfers
    - Takes care of numerous details so you don't have to
    - Uses best practices by default
  - Advanced performance and reliability features
  - Saves you time and effort!
    - Reduces transfer time
    - Reduces learning curve
    - Reduces manual transfer management

Question? Use the Webex chat facility to ask the Host

Shift Reductions in Workflow Execution Time



Question? Use the Webex chat facility to ask the Host

Review of Basic Shift Usage

Shift Client Setup

- NAS HEC front-ends (i.e. pfe/bridge/lfe)
  - None!
  - Already exists as "shiftc" in /usr/local/bin
- Remote hosts (non-NAS or NAS non-HEC)
  - Operates via the Secure Unattended Proxy (SUP)➔➔➔

Question? Use the Webex chat facility to ask the Host

Quick Secure Unattended Proxy (SUP) Review

- What is it?
  - Authentication/authorization mechanism (peer to SFEs)
  - Allows specific transfer/other commands to be invoked on HEC front-ends without SecurID for up to a week
  - Supports direct remote transfers without intermediate bottlenecks
- Usage
  - Download client (wget -O sup http://www.nas.nasa.gov/hecc/support/kb/file/9)
  - Make client executable (chmod 700 sup)
  - Move client to directory in your $PATH (mv sup ~/bin)
  - Authorize host for SUP operations (ssh pfe touch ~/.meshrc)
  - Authorize directories for writes (ssh pfe echo /nobackup/user >> ~/.meshrc)
  - Prepend "sup" to normal commands (sup scp file pfe:/nobackup/user)
- More information
  - Previous webinar "Simple Automated File Transfers Using SUP and Shift"
  - http://www.nas.nasa.gov/hecc/support/kb/entry/145

Question? Use the Webex chat facility to ask the Host

Shift Client Setup (cont.)

- NAS HEC front-ends (i.e. pfe/bridge/lfe)
  - None!
  - Already exists as "shiftc" in /usr/local/bin
- Remote hosts (non-NAS or NAS non-HEC)
  - Operates via the Secure Unattended Proxy (SUP)
  - ➔➔➔Install SUP client if not already done
  - Shift embedded within client using "sup shiftc ..."
  - Need to authorize NAS HEC hosts for SUP operations
    - Create ~/.meshrc if it does not exist (on NAS HEC host)
  - Need to authorize directories for writes
    - Add top level directories to ~/.meshrc (on NAS HEC host)

Question? Use the Webex chat facility to ask the Host

Shift Transfer Initialization

- Local transfers (just like "cp")
  - **bridge%** cp /file1 /file2
  - **bridge%** shiftc /file1 /file2
- Intra-enclave transfers (just like "scp")
  - **bridge%** scp /file1 lou:/file2
  - **bridge%** shiftc /file1 lou:/file2
- Remote transfers (just like "sup scp")
  - **yourhost%** sup scp /file1 pfe:/file2
  - **yourhost%** sup shiftc /file1 pfe:/file2
  - Use "sup -u NAS_user" if remote/NAS username differs
  - Can use balancer aliases (i.e. pfe/bridge/lfe) and unqualified NAS hosts from anywhere!

Question? Use the Webex chat facility to ask the Host

Shift Transfer Initialization (cont.)

- Common initialization options
  - Recursive transfers (-r/-R/--recursive...just like cp/scp)
    - Copy directories recursively
  - Attribute preservation (-p/--preserve...just like cp/scp) (now the default!)
    - Preserve times, permissions, and ownership
  - Symbolic link dereferencing (more on this later!)
    - Never follow links (-P/--no-dereference...just like cp)
    - Always follow links (-L/--dereference...just like cp)
  - Directory handling
    - Create missing parent directories (-d/--directory...just like install)
    - Treat target as a file (-T/--no-target-directory...just like cp/install)
  - Data encryption (--encrypt)
    - Encrypt data stream(s) during remote transfers
    - Eliminates bbftp so may reduce performance in some cases

Question? Use the Webex chat facility to ask the Host

Shift Transfer Initialization (cont.)

- Shift computes file operations in transfer and prints transfer id
  - Initialization output
    - Directories/files found: 0/1
    - Shift id is 1
  - The id can be used to manage/monitor a particular transfer
- After initialization, Shift detaches and begins the transfer
  - You do not need to stay logged on to the origin system!
  - You will be notified by email of completion, errors, and/or warnings
- Never ever remove source files before Shift reports "done"!
  - Wait for completion email or check status with "shiftc --status"
  - Shift may process files in a different order than other tools
    - The existence of a particular file does not imply the existence of others
  - Shift may operate on different portions of the same file at the same time
    - A file may show up with full size but still be incomplete

Question? Use the Webex chat facility to ask the Host

Transfer Management/Monitoring

- A running transfer may be stopped at any time from any host
  - shiftc --stop --id=N
  - Batches of file operations in progress will run to completion!
- A stopped/failed transfer (state = stop/error) may be restarted
  - shiftc --restart --id=N
  - Completed operations will not be run again
  - Failed/unattempted operations will be retried/attempted
  - This is the best and fastest way to recover from transient errors!
  - Must restart on original host or one with equivalent file system access!
- Shift provides history and status of transfers
  - shiftc --history
  - shiftc --status
  - Transfer data only kept for one week after completion/error/stop!

Transfer History

```
pfe% shiftc --history

    id | origin             | command
    ---+--------------------+--------------------------------------
     1 | pfe1.nas.nasa.gov | shiftc file1 /tmp/dir1
     2 | pfe1.nas.nasa.gov | shiftc -p file1 cfe2:
     3 | your_localhost    | sup shiftc -r --verify /tmp/dir1 cfe2:/tmp/dir2
     4 | your_localhost    | sup shiftc -r --encrypt cfe2:/tmp/dir2 .
     5 | pfe1.nas.nasa.gov | shiftc -r --hosts=4 bigdir1 /nobackup/user1/bigdir2
```

Question? Use the Webex chat facility to ask the Host

Transfer Status

```
pfe% shiftc --status

id | state | dirs | files |     file size | start | time |     rate
   |       | sums | attrs |      sum size |       |      |
---+-------+------+-------+---------------+-------+------+---------
 1 | done  |  0/0 |   1/1 |     92KB/92KB | 10/03 |   2s |   46KB/s
   |       |  0/0 |   0/0 |     0.0B/0.0B | 17:06 |      |
 2 | done  |  0/0 |   1/1 |     92KB/92KB | 10/03 |   8s | 11.5KB/s
   |       |  0/0 |   1/1 |     0.0B/0.0B | 17:06 |      |
 3 | done  |  1/1 |   2/2 |     99KB/99KB | 10/03 |   1s |   99KB/s
   |       |  4/4 |   0/0 |   198KB/198KB | 17:07 |      |
 4 | error |  1/1 |   1/2 |     92KB/99KB | 10/03 |   3s | 30.7KB/s
   |       |  0/0 |   0/0 |     0.0B/0.0B | 17:08 |      |
 5 | done  |  1/1 | 64/64 | 65.5GB/65.5GB | 10/03 |  29s | 2.26GB/s
   |       |  0/0 |   0/0 |     0.0B/0.0B | 17:09 |      |
```

Question? Use the Webex chat facility to ask the Host

Detailed Transfer Status

```
yourhost% sup shiftc --status --id=2

    state | op      | target                | size | start | time |   rate
          | tool    | message               |      |       |      |
    ------+---------+-----------------------+------+-------+------+-------
    done  | cp      | cfe2:/u/user1/file1 | 92KB | 10/03 |   5s | 18KB/s
          | bbftp   | -                     |      | 17:06 |      |
    done  | chattr  | cfe2:/u/user1/file1 |    - | 10/03 |   1s |      -
          | sftp    | -                     |      | 17:06 |      |

yourhost% sup shiftc --status --id=4 --state=error

    state | op     | target               | size | start | time | rate
          | tool   | message              |      |       |      |
    ------+--------+----------------------+------+-------+------+-----
    error | cp     | /tmp/dir2/file2      |  7KB |     - |    - |    -
          | rsync  | rsync: send_files    |      |       |      |
          |        | failed to open:      |      |       |      |
          |        | Permission denied    |      |       |      |
```

Question? Use the Webex chat facility to ask the Host

More Information on Basic Usage

- Previous webinar
  - "Simple Automated File Transfers Using SUP and Shift"
  - http://www.nas.nasa.gov/hecc/support/past_webinars.html
- Knowledge base article
  - http://www.nas.nasa.gov/hecc/support/kb/entry/300

New Shift Features and Advanced Shift Usage

Tar Creation/Extraction

- Tar is useful for consolidating many files into one
- But...tar is sloooooooowww!
- Shift now supports tar creation and extraction
  - Transfer to/from tar file instead of transfer to/from directory
  - Uses high-speed transports underneath
  - Supports all standard Shift options (e.g. verification, parallelization, ...)
- Tar creation
  - shiftc --create-tar /some/dir /some/file dirfile.tar
- Tar extraction
  - shiftc --extract-tar dirfile1.tar dirfile2.tar /some/dir
- File size does not necessarily indicate completion status
  - Always wait until Shift reports "done"!

Question? Use the Webex chat facility to ask the Host

Remote Tar Creation/Extraction

- Like normal Shift transfers, source(s) or target may be on a remote host (not both!)
- Remote tar creation with local files
  - shiftc --create-tar /some/dir /some/file lfe:dirfile.tar
- Local tar creation with remote and local files
  - shiftc --create-tar lfe:/some/dir /some/file dirfile.tar
- Remote tar extraction from local files
  - shiftc --extract-tar dirfile1.tar dirfile2.tar lfe:/some/dir
- Local tar extraction with remote and local files
  - shiftc --extract-tar lfe:dirfile1.tar dirfile2.tar /some/dir

Remote Tar vs. Local Tar + Remote Transfer

- Remote tar
  - Inefficient network write
  - Less inefficiency as file sizes increase
  - No additional quota consumed
- Local tar + remote transfer
  - Inefficient local write + efficient local read + efficient network write
  - Local write can be highly parallelized
  - Consumes additional quota
- Performance difference
  - Assuming parallelized creation and non-parallel transfer
    - LAN: 2-2.5x faster using local tar + remote transfer
    - WAN: 1.5-2x faster using local tar + remote transfer
    - Much closer if parallel hosts not available for tar creation
  - Assuming parallel transfer and larger files
    - Transfer cost of both will be similar but remote tar has near zero creation cost

Question? Use the Webex chat facility to ask the Host

Creating Split Tar Files

- Tar files beyond a certain size become detrimental to some file systems (e.g. 1 TB for DMF)
- Shift can split tar files at "around" a given size
  - shiftc --create-tar --split-tar=100m dir dir.tar
  - Use m/g/t for MB/GB/TB
- First split will ALWAYS be called the original name!
- Extra splits will be called name.tar-i.tar for i ≥ 1
  - All files associated with "name.tar" will begin with "name.tar"
- Split files may be as large as the given size plus the size of the largest file being tarred
- This functionality is enabled by default (at 500 GB)!
  - Disable with --split-tar=0

Question? Use the Webex chat facility to ask the Host

Extracting Split Tar Files

- Shift does not have special handling for split tar files
  - Each individual split may be extracted independently
  - Use wildcard "name.*tar" to extract all splits at once
- To extract all local splits at once
  - shiftc --extract-tar name.*tar dir
- To extract all remote splits at once
  - You must quote remote path(s) to preserve the wildcards
  - Intra-enclave
    - shiftc --extract-tar lfe:'name.*tar' dir
  - Non-NAS
    - sup shiftc --extract-tar pfe:'/nobackup/user1/name.*tar' dir

Question? Use the Webex chat facility to ask the Host

Creating Tar Table of Contents

- On DMF file systems (e.g. Lou home file system), large tar files will eventually be migrated to tape
- By default, there's no way to see the contents of a migrated tar file without first retrieving from tape
- Shift can create associated table of contents file(s)
  - Contains "ls -l" output of tar file(s) (like "tar -tv")
  - Much smaller than tar files so not typically migrated
  - Works with both local and remote tar files
  - shiftc --create-tar --index-tar dir dir.tar
    - Creates file "dir.tar.toc"
  - shiftc --create-tar --index-tar --split-tar=100m dir dir.tar
    - Creates "dir.tar.toc" and "dir.tar-i.tar.toc" for each split i

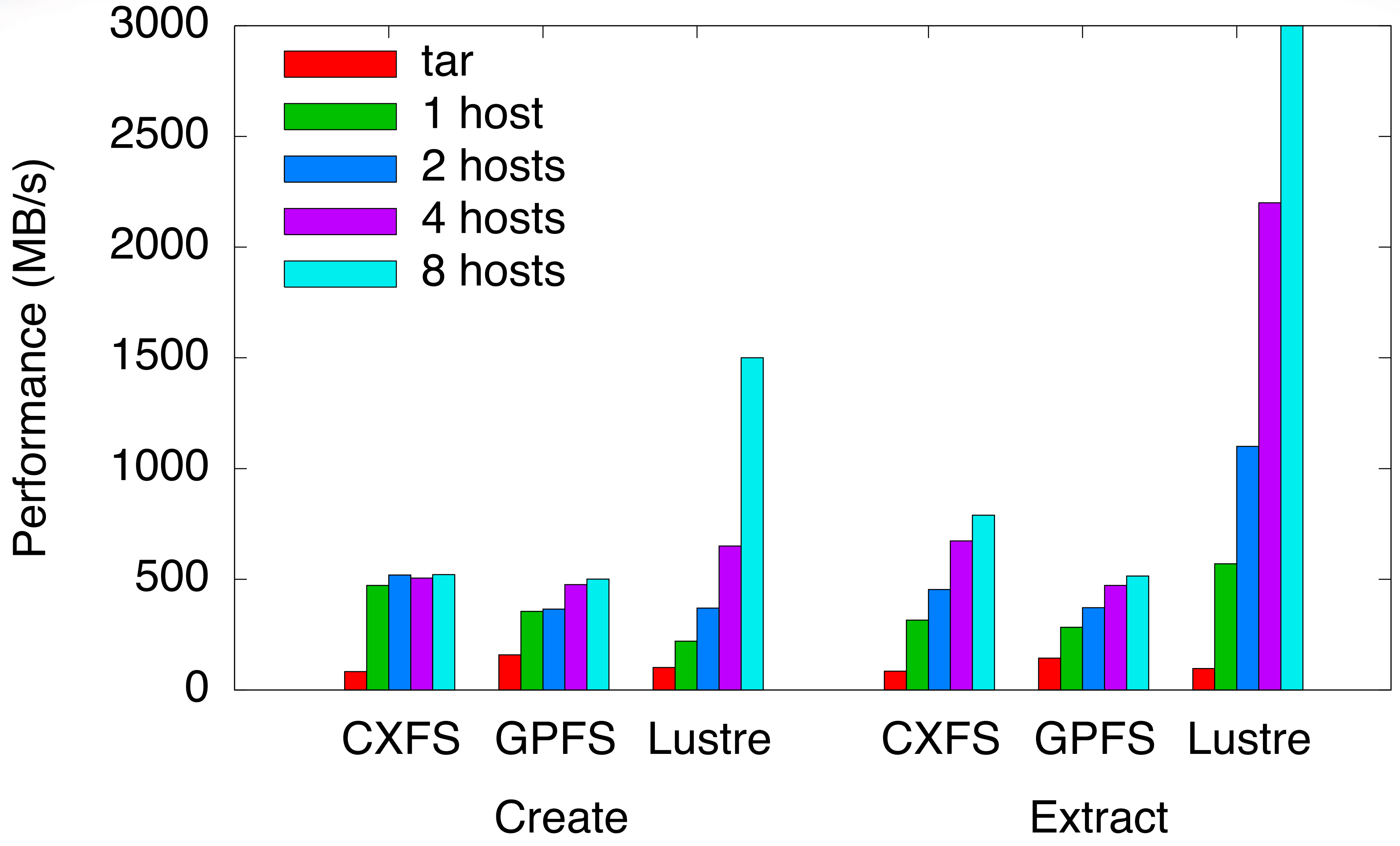
Question? Use the Webex chat facility to ask the Host

Tar Functionality Limitations

- Does not handle compressed tar files
- Currently only supports POSIX ustar tar format
  - Symbolic links of at most 100 characters
  - Pathnames divisible by "/" into a 155 character max prefix and 100 character max trailing path
  - Shift does support extensions for large files sizes, uids, and gids standard in most tar versions
  - Tars produced by OSX/BSD tar not in ustar by default
    - On NAS OSX systems, can use /usr/bin/gnutar

Question? Use the Webex chat facility to ask the Host

Shift Local Tar Performance (64*1GB Files - measured with initial prototype)



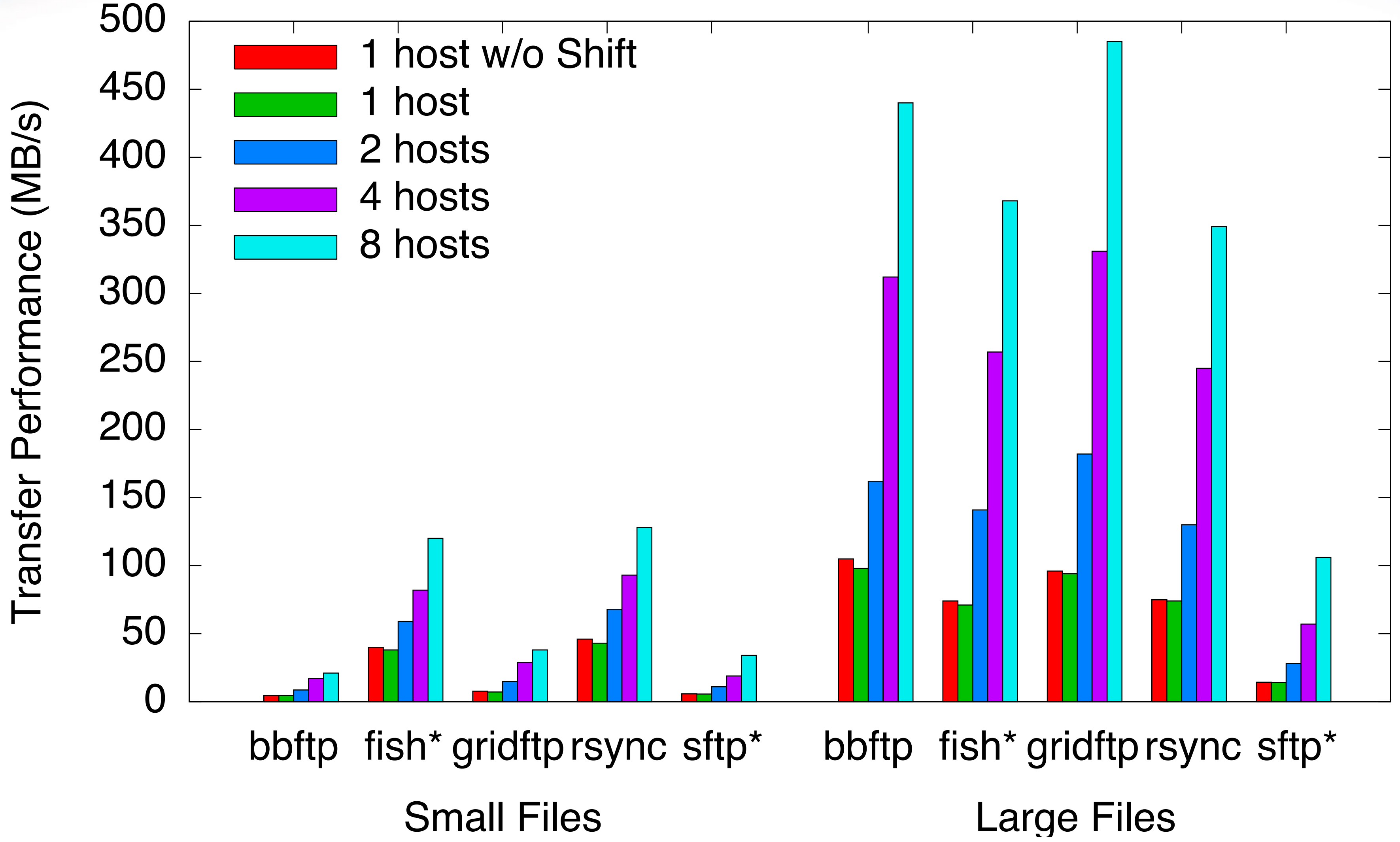
Question? Use the Webex chat facility to ask the Host

Host Parallelization

- Shift can still utilize multiple hosts to parallelize a single transfer using --hosts
- This is the easiest way to significantly improve transfer performance within NAS HEC enclave!
  - shiftc --hosts=8 --create-tar dir /nobackupp3/user/dir.tar
  - shiftc --hosts=2 -r /nobackupp3/user/dir lfe:
- Remote host parallelization only works if you have multiple hosts at your site with access to the same src/dst file system!
  - sup shiftc --extract-tar --hosts=4 pfe:/nobackup/user/dir.tar .
  - You must let Shift know about your hosts
    - Run a small remote transfer from each host to authorized directory
      - touch foobar; sup shiftc foobar pfe:/nobackup/user
  - If your hosts require pubkey authentication to move between them
    - Before transfer initialization, have ssh agent running with valid key

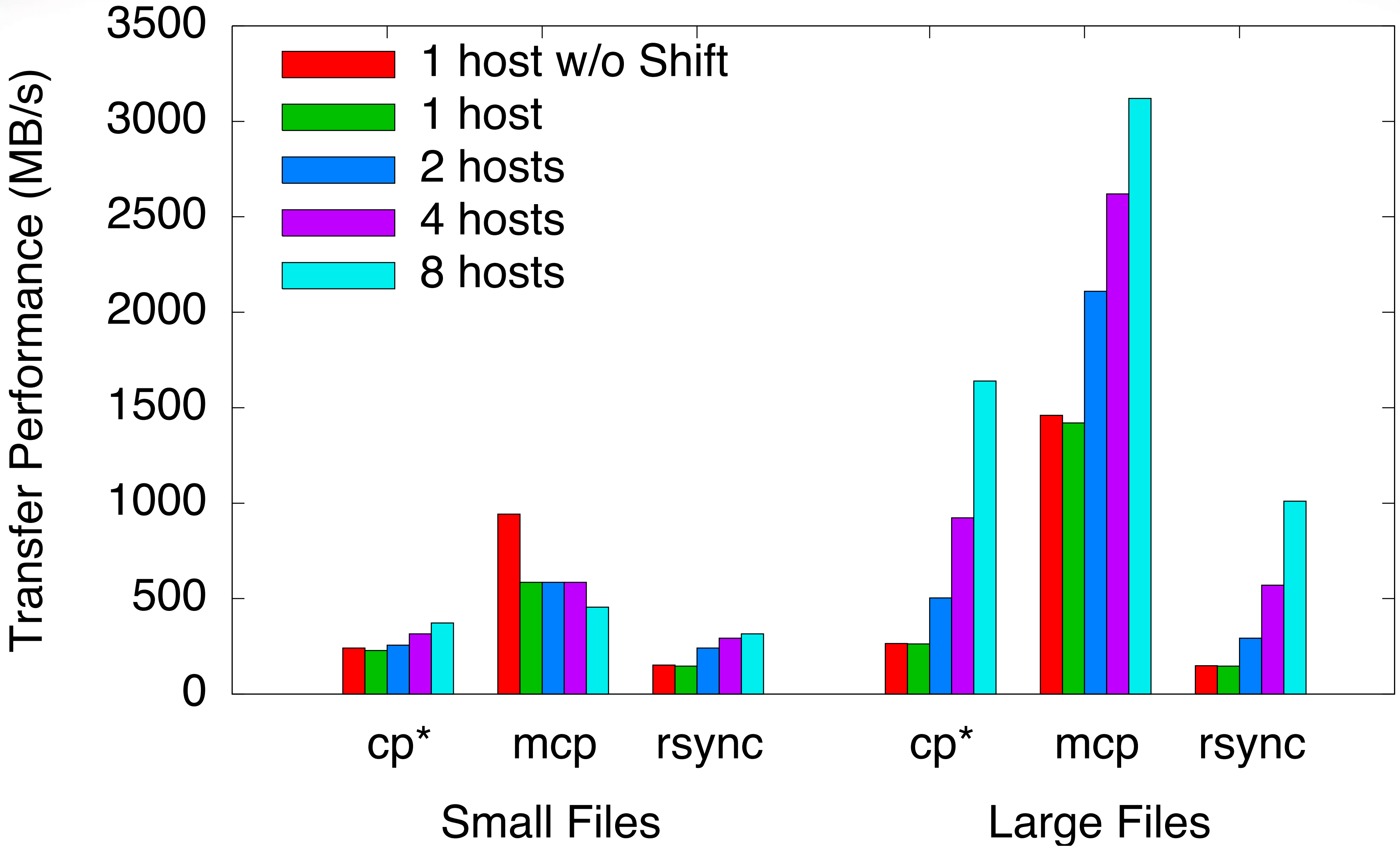
Question? Use the Webex chat facility to ask the Host

Shift Remote Performance (1k*4MB Files and 64*1GB Files via 10 GE WAN)



Question? Use the Webex chat facility to ask the Host

Shift Local Performance (1k*4MB Files and 64*1GB Files Lustre->Lustre)



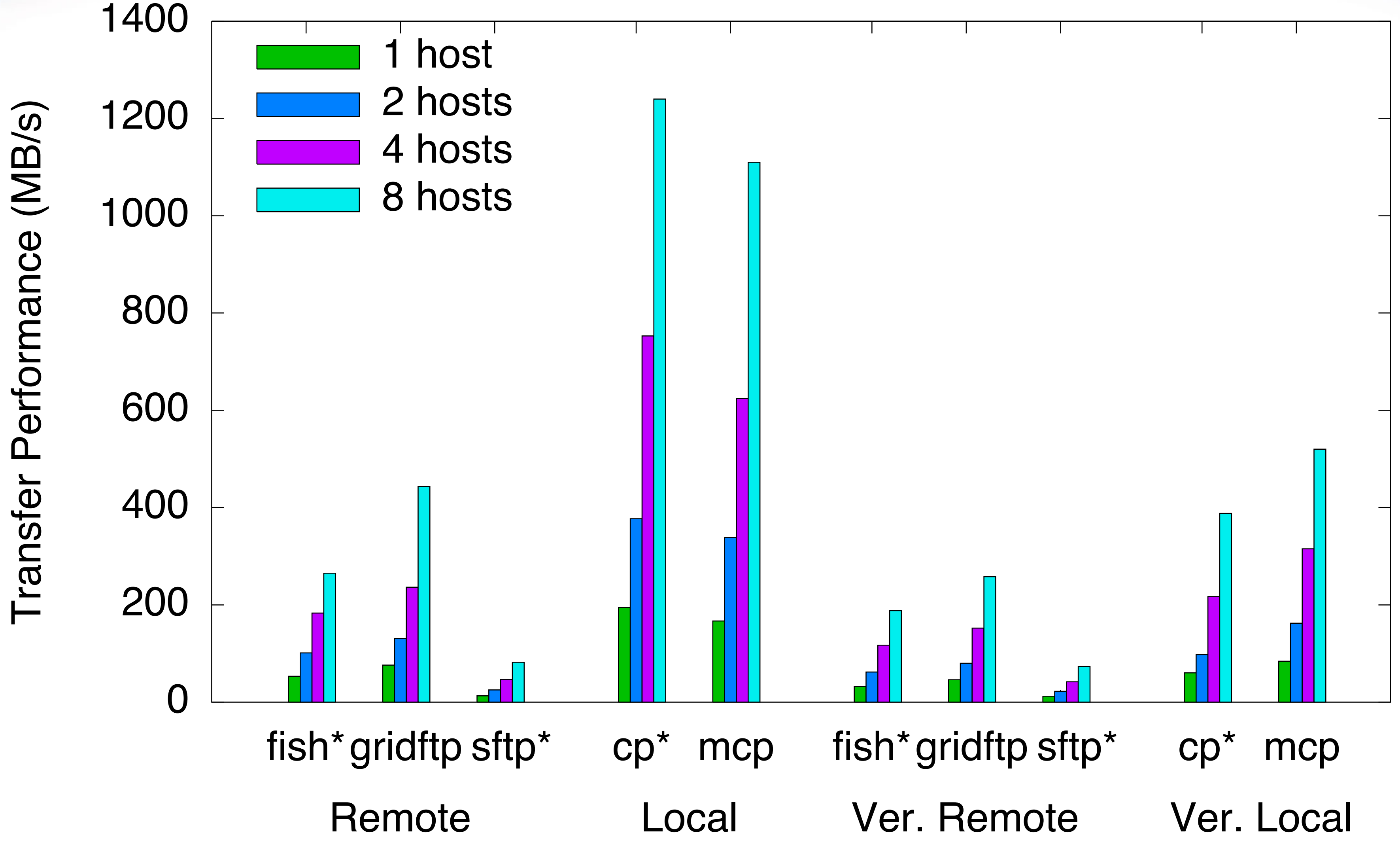
Question? Use the Webex chat facility to ask the Host

Balancing Parallel Workload

- Parallelization achieves highest performance when all participants finish their workload at the same time
  - Large enough granularity to minimize various overhead
  - Small enough granularity so faster participants can make up for slower
- Transfers of large files can cause workload imbalances
  - The --split option allows large files to be split into multiple workloads
    - sup shiftc --split=1g --hosts=8 /big/file bridge:/dir
  - The --files and --size options allow further tuning of max files processed in each batch and max size of each batch
    - sup shiftc --split=1g --files=20k --size=8g --hosts=8 pfe:/big/dir .
- Previously, built-in remote transport too slow to be worthwhile
  - New built-in remote transport ("fish") over 5x faster now makes practical

Question? Use the Webex chat facility to ask the Host

Shift Single File Parallelization Performance (1 64GB File via 10 GE WAN and Lustre->Lustre)



Question? Use the Webex chat facility to ask the Host

Client Parallelization

- Shift can now also use multiple clients on the same host to parallelize a single transfer using --clients
  - shiftc --clients=2 --split=1g -r /nobackupp3/user/dir lfe:
- Only useful when single client cannot consume all resources of a single host
  - Remote transfers via ssh-based transports (rsync, fish, sftp) where single client may not consume full network bandwidth
  - Transfers with --verify where CPU-intensive hashes may overlap with I/O intensive data transfer
- Initial performance measurements (64*1GB files)
  - local: 36% faster with 2 clients, 52% faster with 4 clients
  - verified local: 53% faster with 2 clients, 64% faster with 4 clients
  - LAN: 87% faster with 2 clients, 3.1x faster with 4 clients
  - verified LAN: 62% faster with 2 clients, 2.7x faster with 4 clients

Question? Use the Webex chat facility to ask the Host

Multiple Independent Transfers

- Running many transfers at once can overwhelm the resources of a system or even the whole computing environment
- Shift now has the ability to wait until completion
  - Can script multiple transfers to follow each other without overload
  - shiftc --wait /some/file /some/dir
  - Exits with 1 for failure/interruption or 0 for success
  - Status emails disabled; instead prints transfer summary upon completion
- Shift can also link multiple transfers together under a single transfer id
  - Arbitrary source/target lines read from stdin
  - echo "/file1 /dir1" > in; echo "/file2 /dir2" >> in; shiftc < in
  - Command line options apply to all inputs (e.g. -r, --hosts, etc.)
  - Resource usage is equivalent to that of a single transfer

Question? Use the Webex chat facility to ask the Host

New Integrity Verification Features

- Shift can still verify that file contents on destination disk match contents on source disk using --verify
  - shiftc --verify /some/file pfe:/dir
- Shift can now also verify integrity of tar creations/extractions
  - Contents of files within archive summed against those on disk
  - During creation, additionally validates tar headers, padding, etc.
  - Corruption will be automatically corrected when feasible
- Shift now has a default lightweight validation mechanism even when --verify not used
  - Checks existence and sizes of destination files after transfer
    - Provides assurance that files were not inappropriately marked done
    - Run during attribute preservation phase (-p/--preserve now default)
    - Error messages about differing src/dst file sizes are important!
  - Continue to use --verify if you wish to verify actual file contents!

Question? Use the Webex chat facility to ask the Host

File/Directory Synchronization

- Shift could previously (and still can) do some synchronization using --local=rsync or --remote=rsync
  - No ability to control how synchronization carried out
  - Not able to take advantage of higher speed transports or hashing capabilities
- Shift now has a native synchronization mechanism
  - shiftc --sync -r /dir1 /dir2
  - Files with same modification time and size are skipped by default
  - Non-existent files are transferred and verified
  - Existing files are verified and reconciled when necessary
- Due to implementation details, synchronized transfers will show warnings (state = "run+warn") when portions of files need to be copied
  - Ignore these warnings unless they turn into errors!
- Initial performance measurements
  - 40% faster than rsync for 64*1GB files and 1 byte changed in 2 files
  - 2.2x faster than rsync for 1k*4MB files with modification times changed
  - 6.5x faster than rsync for 1*64GB file with 1 byte changed using 8 hosts

Question? Use the Webex chat facility to ask the Host

Synchronization Usage Notes

- Can skip modification time and size checks using -I/--ignore-times (just like rsync) to force full checksum verification
- Target follows normal -r semantics
  - Source placed beneath target if target exists, renamed to target otherwise
  - To synchronize directory to same name, specify parent of desired destination
    - shiftc --sync -r /dir1 /dir1_parent (to sync /dir1 with /dir1_parent/dir1)
  - To synchronize to different name, use -T/--no-target-directory
    - shiftc --sync -r -T /dir1 /dir2 (to sync /dir1 with /dir2)
- Can be used for after-the-fact verification
  - shiftc -r dir1 dir2 (oops...forgot --verify)
  - shiftc --sync --ignore-times -r dir1 dir2 (now verified)
- Synchronizing to DMF file system can be time-consuming
  - Files may need to be retrieved from tape to checksum
- Mutually exclusive with --create-tar and --extract-tar
  - Not practical to change sizes of files within tar archives

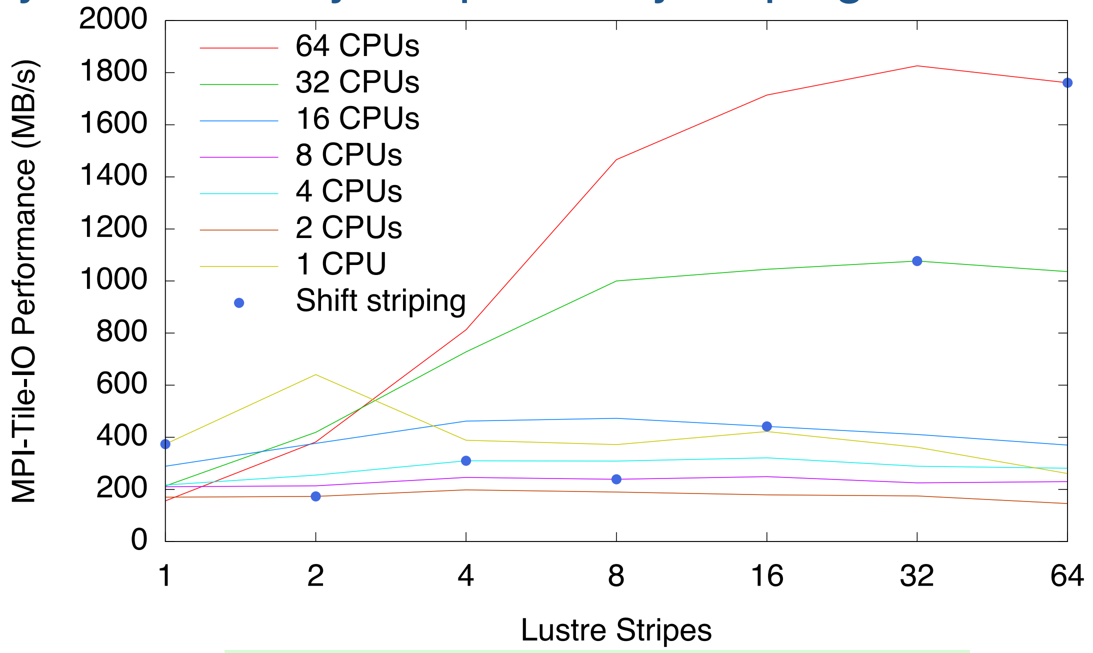
Question? Use the Webex chat facility to ask the Host

Revamped Retry Mechanism

- Previous mechanism was based on a time interval in which operations would be attempted multiple times
  - Eventually became a noop (besides rectification of corruption) due to incompatibilities with other features that were added
- New mechanism based on fixed number of retries per file
  - shiftc --retry=3 /some/file /some/dir
  - Default number of retries is 2
  - Disable retries by setting to 0 (do not disable retries with --sync!)
- A host performs exponential backoff if all operations in a batch fail to give time for transient errors to correct themselves
- Transfer state shows "run+warn" when operations undergoing retry
- New mechanism should help with most common transient failure
  - bbftp port exhaustion due to inefficient port usage

Automatic Lustre Striping (New behind the scenes)

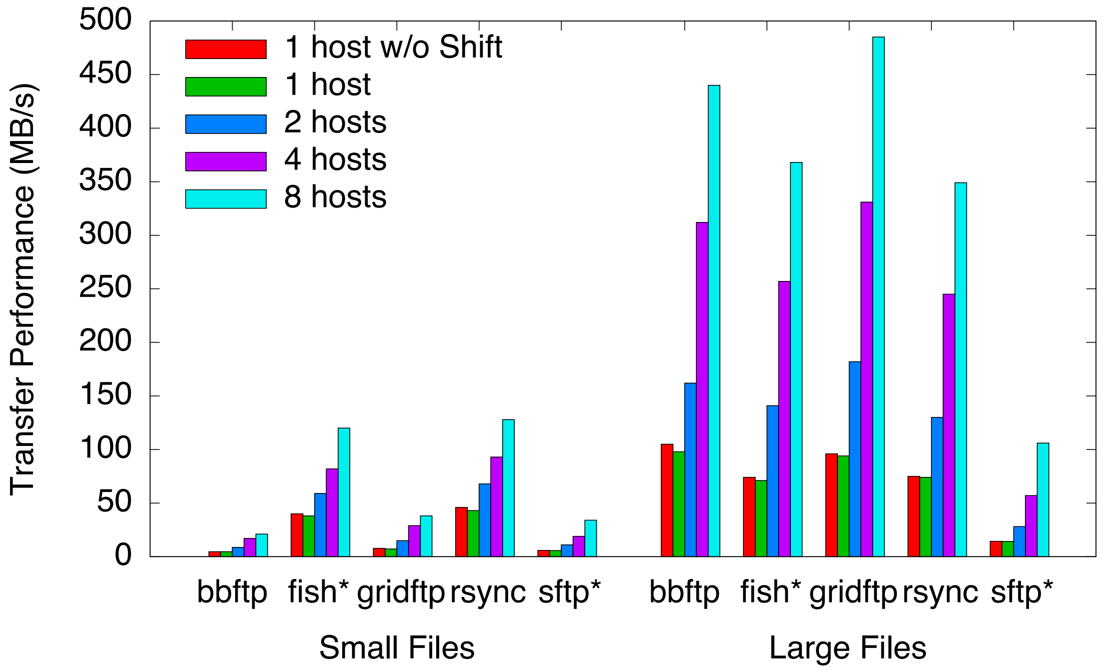
- Files transferred to Lustre are striped based on size
  - Uses 1 stripe per GB
  - Covers standard transfers, tar creations, and tar extractions
- Does not work in all scenarios
  - Striping cannot be changed when overwriting existing files
  - Temporary files used by bbftp destroy striping



Question? Use the Webex chat facility to ask the Host

Dynamic Transport Optimization (New behind the scenes...since last webinar)

- The relative performance between different transports varies depending of file size
- Shift adjusts the transport used for each batch of files to maximize performance (subject to availability at src/dst)
  - You may see different transports used within the same transfer when looking at its detailed status



Question? Use the Webex chat facility to ask the Host

Symbolic Link Dereferencing

- An old feature, but important consideration for transfers is deciding how to handle symbolic links
  - There is frequent collaboration at NAS with many links to other user files
  - Archived data (tape or tar) should allow results to be reproduced
- Default behavior (just like cp)
  - Follow file links, but do not follow directory links
  - Will cause unrecoverable errors when referenced files do not exist
- Always follow links using -L/--dereference (just like cp)
  - Can result in duplicate files at destination
  - Useful if need full data set from other users for archival purposes
  - Will cause unrecoverable errors when referenced files/dirs do not exist
- Never follow links using -P/--no-dereference (just like cp)
  - Can result in broken links at destination
  - Will always succeed now...but could be missing critical data later

Conclusion

- Summary of new features
  - Local and remote tar creation/extraction with splitting, indexing, verification, parallelization
    - Measured create at 15x faster than tar, extract at 30x faster
  - New built-in remote transport makes parallelization of single file remote transfers practical
    - 8 host remote transfer of single 128 GB file at 434 MB/s with peaks of 680 MB/s
  - Client parallelization to squeeze every last resource out of each host
    - Local transfer 36% faster and LAN transfer 87% faster with 2 clients
  - Blocking transfers for scripting multiple transfers without overloading systems/environment
  - Lightweight existence/size checks provide additional assurance of successful file transfer
  - Native synchronization mechanism based on high-speed transports
    - 8 host sync of single 64 GB file at 6.5x faster than rsync
  - Revamped retry mechanism to more successfully overcome transient failures
  - Automatic Lustre striping transparently optimizes later I/O access by parallel jobs
  - Dynamic transport optimization uses the fastest transport for each batch of files
- You've hopefully learned
  - How to utilize these features effectively
  - Other tips and tricks to get the most out of Shift

Question? Use the Webex chat facility to ask the Host